Technical Cybersecurity

ROP OPs

How ROP Works

MACHINE CODE IS DENSE

- Binary strings
- Meaning depends on where you start

returnorientedprogrammingisthebesticantgetenoughofit

How ROP Works

returnorientedprogrammingisthebesticantgetenoughofit

INTENDED

- Return oriented programming is the best, I can’t get enough of it.

WHAT ELSE CAN IT SAY?

- Iamthebest: I am the best
- hecantturnrogue: He can’t turn rogue
- …you get the idea.

How ROP Works

returnorientedprogrammingisthebesticantgetenoughofit

WE ASSEMBLE FROM GROUPS OF POINTERS

- Iamthebest: I am the best
  - Character positions: 25, 19, 27, 30
- We can assemble this sentence with pointers to character positions
  - We may have multiple options too, as ‘i’ is in many places

ROP is similar

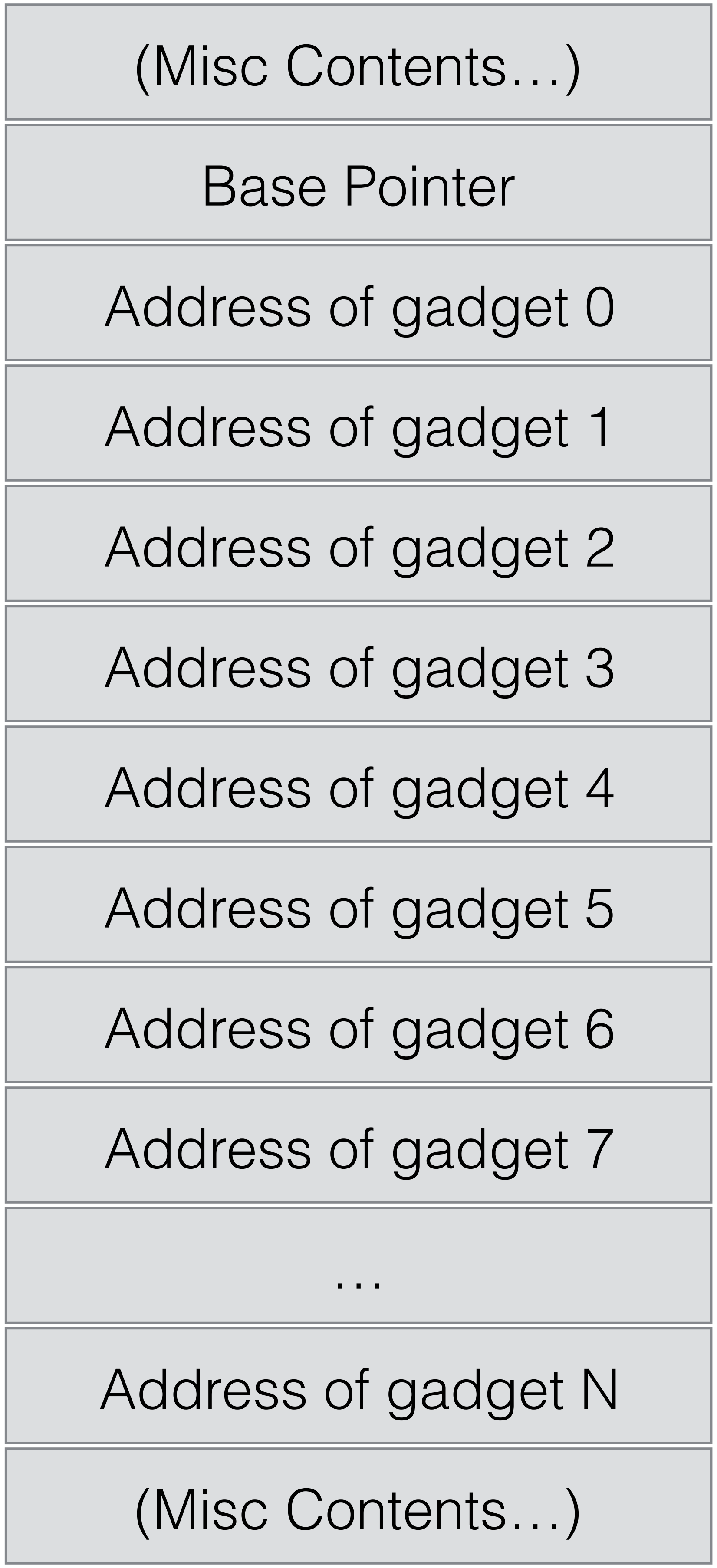
WE LOOK FOR AN INSTRUCTION FOLLOWED BY A RET

- RET: Return from procedure
  - When called without an argument will pop an address from the stack and place that address into EIP/RIP
- usually called in a program after a CALL instruction, but we won’t be using it that way
- Each instruction/RET sequence is called a **gadget**
- We string these gadgets into a **ROP Chain**

ROP Stack

STACK LOOKS LIKE THIS

- The RET at the end of a gadget will start execution at the address popped from the stack



(Misc Contents…)

Base Pointer

Address of gadget 0

Address of gadget 1

Address of gadget 2

Address of gadget 3

Address of gadget 4

Address of gadget 5

Address of gadget 6

Address of gadget 7

…

Address of gadget N

(Misc Contents…)

ROPs are difficult and
have lots of moving parts.